Attorney's Docket No.: 17646-010002 / 20000016-DIV

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Larry D. Hebel, et al
Serial No. : 09/532,404
Filed : March 22, 2000
Title : DYNAMIC METHOD FOR CONNECTING A CLIENT TO A SERVER APPLICATION

Art Unit : 2141
Examiner : Quang Nguyen, Ph.D.

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF ON APPEAL

(1) Real Party in Interest

Computer Associates Think, Inc., the assignee of the present Application, is the real party in interest.

(2) Related Appeals and Interferences

There are no related appeals or interferences.

(3) Status of Claims

Claims 7-10 and 16-36 were pending in the application, with claims 7, 16, 24, and 32 being independent. All pending claims stand rejected.

(4) Status of Amendments

No response to the final Office Action dated August 10, 2004 was filed and no amendments are being submitted herewith.

(5) Summary of Claimed Subject Matter

A project sharing system according to one preferred embodiment includes client workstations with project sharing enabled and a project server. A project client workstation (sometimes referred to as client referring to the software on the workstation machine) knows

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 325808406 US

March 10, 2005
Date of Deposit

how to coordinate with other project clients workstations through the use of the project server. The project server is a separate program that coordinates project clients at client workstations. It offers no user interface and is only useful to project client at client workstations. The server is connected to the workstations by a communications link including a hardware connection link like a LAN and controls such as TCP/IP.

Client workstation can only communicate to a server, which in turn can only communicate to client workstations. No direct messaging occurs between client workstations over system. Messages will always be routed through the server. Servers do not interact with other servers as each controls a different project database. Further, a client workstation using system can only connect to one server at any one time, where a server can connect to many client workstations, allowing the server to broadcast to all client workstations. This type of communications configuration is sometimes known as a hub (or star). The client and the server software are distinct pieces of software and may be executed on the same or different machines. There is no dependence on any specific piece of hardware. When the client software is on a machine it is termed herein a client workstation and when a server software is on a machine it is a server. The server will continue running until all client workstations have disconnected from it.

In particular embodiments, the system uses the Berkeley compatible TCP/IP to requires programs to have a 32-bit IP address and a 16-bit port number in order to provide connectivity. Transmission Control Protocol/Internet Protocol (TCP/IP) is a well known standard where TCP controls the data transfer and IP provides the routing through hardware connections between client workstations and servers. IP addresses resolve machine locations, and port numbers are used to resolve client and server process locations on the client workstation. At any one instance, the combined IP address and port number may be used to uniquely identify any client workstation or server application. TCP/IP stream sockets receive messages in the order they are sent. They do not inherently provide a mechanism for controlling the order in which messages are processed. Not all messages are weighted equally in importance or in the amount of processing required. In addition, the server will be managing many sockets (one per client). As the activity to the server increases, the rate of event handling inevitably lags behind. When this happens, some events may become invalid prior to processing. The system uses a queuing mechanism to provide what TCP/IP doesn't, a priority messaging system where high priority messages can supersede low priority messages.

The client workstations and server typically have three stages of operation — startup, event handling, and shutdown. The server, on startup, will query the host machine 's IP address and write both the IP address and the user supplied port number into the database's access log file. The client workstations on startup, will read the server specific IP address and port number from the same file. This is necessary for two reasons: i) there is only one server per database and any attempt to start a subsequent server for the same project would fail, because the file is being accessed by the initial server; and ii) this allows the client workstations to find the server, since the user can start the server on any workstation machine.

Priority messages handling scheme will ensure two features. One is no client workstation will starve from lack of server attention. The second is that messages received by the client workstations and server at any instance will be handled from highest to lowest priority. Starvation is avoided using a rotation scheme. All awaiting messages are moved to a queue before they are processed. This buffering of incoming messages provides the basis of priority messaging. Received messages are insertion sorted into the queue by priority. For the server, after all waiting client messages have been read the messages in a queue Q can be selectively handled. Any messages arriving while messages are being processed are not moved to the queue until it has been emptied at which time the next rotation occurs.

For example, assume the messages reach the server in the order A, B, and C with priorities 2, 1, and 2 respectively. Message B with priority 1 will get handled first. A and C will then follow in that order since equal priority messages are handled by rotation order. Note that even if message C were to reach the server before message A, message A would still be handled first, because it has precedence in the rotation. However, if message D, priority 1 shows up while the handling for messages A, B, and C have already started, it will have to wait until the next rotation.

(6) Issues

A) Would Claims 7-10 and 16-18 have been obvious, under 35 U.S.C. § 103(a), over U.S. Patent No. 5,699,523 to Li *et al.* (hereinafter "*Li*") in view of U.S. Patent No. 5,231,633 to Hluchyj, *et al.* (hereinafter "*Hluchyj*").

B) Would Claims 19-36 have been obvious, under 35 U.S.C. § 103(a), over *Li* in view of *Hluchyj* and further in view of U.S. Patent No. 5,179,708 to Gyllstrom *et al.* (hereinafter "*Gyllstrom*").

(7) Grouping of Claims

The claims stand or fall together. In so grouping the claims, Applicants do not admit that the subject matter of the dependent claims represents merely obvious variations under 35 U.S.C. § 103(a) over the subject matter of the respective independent claims.

(8) Argument

The final Office Action, mailed August 10, 2004, rejected Claims 7-10 and 16-18 under 35 U.S.C. § 103(a) as being unpatentable over *Li* in view of *Hluchyj*. The Office Action rejected Claims 19-36 under 35 U.S.C. § 103(a) as being unpatentable over *Li* in view of *Hluchyj* and further in view of *Gyllstrom*. For at least the reasons set forth below, Applicants traverse the rejections of claims 7-10, 16-18, and 19-36 and respectfully request a reversal of the rejections.

First, Applicants respectfully assert that *Li*, *Hluchyj*, and/or *Gyllstrom* fail to disclose, teach, or suggest at least the end-server processing described by "sending a message having a priority level from the client to the server, the message requesting processing by the server … receiving the message at the server … reading the priority level of the message at the server [and] determining at the server a current client rotation position of the client" as recited, in part, in Claim 7. In contrast, the *Hluchyj* system (what the Office Action equates with "the server") is clearly interposed between end systems, with the first end system communicating packets for use or processing by the second end system – the packets do not request processing by the *Hluchyj* system, which performs the asserted packet multiplexing. For example, *Hluchyj* discloses that in typical systems:

> [b]efore the flow of packets between the end systems begins, a connection (or virtual circuit) is established between them. This connection determines the path (i.e., the nodes and internodal trunks) that the fast packets will follow from end to end. FIG. 2 depicts a switch typically used at an intermediate node, that receives fast packets from one or more input trunks and switches them to one or more output trunks.

Hluchyj, 2:1-8. The *Hluchyj* system then enqueues/dequeues packets for transmission via the internodal trunk. *Hluchyj*, 5:38-42. More specifically, *Hluchyj* teaches that packets intended for other recipients are prioritized, put in queues, and multiplexed at such an internodal trunk for transmission to particular recipients beyond the intermediate nodes and internodal trunks. *See Hluchyj*, 1:6-12; *id.* at 4:26-27; *see also* Office Action at 7. Moreover, *Hluchyj* repeatedly teaches that after enqueueing/dequeueing at the intermediate trunk, the packets are transmitted from the trunk to the expected recipient for subsequent processing. *See*, *e.g.*, *Hluchyj*, 4:14-17; *id.*, 4:26-27; *id.*, 4:61-66; *id.*, 5:38-42; *id.*, 6:57-68; *id.*, 9:32-40. Indeed, *Hluchyj* teaches that the disclosed technique attempts to solve bandwidth problems for multiple traffic types for transmission to multiple recipients along a network trunk. *See Hluchyj, 4:14-17.* In other words, *Hluchyj* teaches that after enqueueing/dequeueing at an intermediate node based on traffic type, the packets are transmitted along the internodal trunk to the recipient for subsequent requested processing. *See, e.g., Hluchyj,* FIGs. 2, 4, and 5.

Similarly to *Hluchyj*, *Li* discloses "[a] device for communication *between* at least one client and at least one server." *Li*, Abstract (emphasis added); *id*, Title. *Li* further discloses that the "present invention relates to a router device *between* a client and a server, the method for using the device, and the use of the device." *Li*, 1:9-11 (emphasis added). While not asserted again Claim 7, Applicants further submit that *Gyllstrom* fails to account for these deficiencies of *Hluchyj* and *Li*. For example, *Gyllstrom* also teaches transmission to the "destination" or "recipient" process by a message-delivery function, which determines the message priority. *See Gyllstrom*, Abstract; *id.*, Title, *id.*, FIG. 4; *id.*, FIG. 5; *see also* Office Action at 5.

In response, the Examiner argues, for example, that "sending a message having a priority level from the client to the server, the message requesting processing by the server," in Claim 7, "can be given broad and reasonable interpreted [*sic*] in light of specification as sending a message *(data/voice packets)* having a priority level *(i.e., having a field indicating the degree of priority)* from the client *(from a sending user/application)* to the server, the message requesting processing *(prioritizing, selectively discarding, multiplexing and transmitting, etc.)* by the server *(i.e., by an information processing server, a proxy server, a router, or an internodal trunk, etc).*" Office Action, pg. 7. Yet Applicants respectfully point out that "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art" according to MPEP §2143.03. Claim 7 recites that the message requests "processing by the server," but nowhere has the Examiner explained or directed Applicants to any teaching that *Hluchyj*'s voice/data packets request processing – including the asserted prioritizing, selectively discarding, multiplexing and transmitting – by the internodal trunk. Indeed, the cited portions seem to teach away from such an interpretation. For example, *Hluchyj* teaches that, within the internodal trunk, "packets within a particular traffic type are selected for transmission through use of a head of line priority service ... a packet discard mechanism ... or both." *Hluchyj*, Abstract. In another example, *Hluchyj* discloses a "fast packet priority queueing, selective discarding, and bandwidth allocation methodology." In one embodiment of this methodology, "fast packets of differing traffic types are prioritized pursuant to differing prioritization methods vis-a-vis one another. The prioritized packets from each group are then multiplexed and transmitted." *Hluchyj*, Summary. In other words, the packets in *Hluchyj* do not request the multiplexing, discarding, or transmitting by the internodal trunk, they are selected for it based on traffic type. Accordingly, the Office Action singularly fails to show that *Hluchyj* (even when combined with *Li* and *Gyllstrom*) teaches the end server processing claimed by "sending a message having a priority level from the client to the server, the message requesting processing by the server ... receiving the message at the server ... reading the priority level of the message at the server [and] determining at the server a current client rotation position of the client" as recited, in part, in amended Claim 7.

In a more specific example, Applicants respectfully assert that neither *Li* nor *Hluchyj,* whether alone or in combination, teach, suggest, or disclose "determining at the server a current client rotation position of the client" as recited, in part, in Claim 7. Prior Office Actions agree that *Li* fails to teach such a limitation, but apparently allege that the traffic type round robin in *Hluchyj* accounts for this deficiency. As described above, *Hluchyj* repeatedly discloses that its packets are routed by traffic type – not "a current client rotation position of the client". *See Hluchyj*, Title; *id.* at Abstract; *id.* at 1:7-12; *id.* at 5:29-33. *Hluchyj* teaches that routing by traffic type attempts to solve bandwidth problems for multiple traffic types for transmission along a network trunk. *See Hluchyj*, 4:14-17. This teaching of *Hluchyj* is further reflected in the claims (*Hluchyj*'s independent Claims 1, 5, 6, 8, 9, 14, 21, and 30 each recite a particular "method of post-switching multiplexing fast packets for differing traffic types"). In short, *Hluchyj* teaches that each packet is queued based on a traffic type of the packet, not "a current client rotation position of the client" as recited, in part, in independent Claim 7.[1] Applicants further submits that *Gyllstrom* fails to account for these deficiencies of *Hluchyj* and *Li*. Regardless, the Examiner has repeatedly failed to show how *Hluchyj*'s traffic type equates with "a current client rotation position of the client."

Instead, the Examiner first argued that "the language of the limitation ... can be given broad and reasonable interpreted *[sic]* in light of the specification." Office Action mailed July 14, 2003 at 6. Yet again Applicants respectfully point out that "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art" according to MPEP §2143.03. Applicants submit that the Office Action is effectively *disregarding* "a current client rotation position of the client" without cited support from the current specification for such an interpretation and in contrast to the plain language of Claim 7. In the next Office Action, the Examiner then argued that "one cannot show non-obviousness by attacking references individually where the rejections are based on combinations of references." Office Action mailed February 3, 2004 at 8 (citations omitted). Yet this curiously ignores Applicants' repeated assertions that "neither *Li* nor *Hluchyj*, whether alone or in combination, teach, suggest, or disclose 'determining at the server a current client rotation position of the client' as recited, in part, in Claim 7." *See, e.g.*, Response filed January 14, 2004 (emphasis added). Moreover, the Examiner still relies on *Hluchyj* to attempt to show this limitation and has yet to provide any such teaching in *Li* or *Gyllstrom* – in fact, the Examiner admitted that *Li* fails to include such a teaching. In short, the Examiner has yet to (and can not) show where *Li*, *Hluchyj*, and *Gyllstrom* (whether alone or in combination) teach, suggest, or disclose "a current client rotation position of the client" as recited in Claim 7.

For at least these reasons, *Li*, *Hluchyj*, and *Gyllstrom*, whether alone or in combination, fail to teach, suggest, or disclose at least "sending a message having a priority level from the client to the server, the message requesting processing by the server … receiving the message at the server … reading the priority level of the message at the server [and] determining at the

[1] Applicants further assert that nowhere does *Hluchyj* appear to discuss referencing the sending node, source, or client to determine placement in the weighted round robin (WRR). Indeed, the source of the packets in *Hluchyj* appears to be irrelevant – each embodiment appears to queue packets in the WRR based on the traffic type. *Hluchyj* clearly fails to teach, suggest, or disclose "a current client rotation position of the client" as recited in Claim 7.

server a current client rotation position of the client" as recited, in part, in Claim 7. For analogous reasons, Applicants respectfully assert that *Li*, *Hluchyj*, and/or *Gyllstrom* fail to teach various limitations of independent Claims 16, 24, and 32. Accordingly, Applicants respectfully request at least a reversal of the rejections – if not the allowance – of independent Claims 7, 16, 24, and 32 and claims depending therefrom.

CONCLUSION

For the reasons advanced above, Appellants respectfully submit that the present claims are allowable over the cited prior art references. Reversal of the obviousness rejection under 35 U.S.C. § 103(a) is respectfully requested. If questions remain regarding the above, please contact the undersigned.

The brief fee of $500 is enclosed, along with a $450 fee for a two-month extension of time. Please apply any other charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: March 10, 2005

Thomas H. Reger II
Reg. No. 47,892

Fish & Richardson P.C.
5000 Bank One Center
1717 Main Street
Dallas, Texas 75201
Telephone: (214) 292-4084
Facsimile: (214) 747-2091

Appendix of Claims

7. A method for communication between a client and a server in a computer network, comprising the steps of:

sending a message having a priority level from the client to the server, the message requesting processing by the server;

receiving the message at the server;

reading the priority level of the message at the server;

determining at the server a current client rotation position of the client; and

inserting the message into a message queue for processing by the server in response to the priority level and the current client rotation position of the client.

8. The method of Claim 7, further comprising the steps of sequentially processing a plurality of messages from the message queue by the server.

9. The method of Claim 8, further comprising the steps of storing incoming messages for insertion into the message queue during the sequential processing of messages by the server.

10. The method of Claim 7, further comprising the steps of:

determining address information for the server by the client; and

creating at the client the message including the address information for the server.

16. A network system for processing messages, comprising:

a plurality of clients operable to generate and communicate messages having one or more priority levels to a server, each message requesting processing by the server; and

the server coupled to the clients, the server operable to receive one or more messages from the clients, to determine a priority level for each message, and to process the messages according to the messages' priority levels and the clients' rotation positions.

17. The network system of Claim 16, wherein the server is further operable to process messages that have different priority levels in order of the different priority levels.

18. The network system of Claim 16, wherein the server is further operable to processes messages that have a same priority level and were received from different clients in order of the different clients' rotation positions.

19. The network system of Claim 16, wherein the server is further operable to receive a first message from a first client and a second message from a second client, to process the first message before the second message if the first message's priority level is higher than the second message's priority level, and to process the first message before the second message if the first and second messages have the same priority level and the first client's rotation position is before the second client's rotation position.

20. The network system of Claim 16, wherein the server is further operable to store the messages in a queue according to the messages' priority levels and the clients' rotation positions and to process the message in order of storage in the queue.

21. The network system of Claim 20, wherein the server is further operable to store messages that have different priority levels in order of the different priority levels.

22. The network system of Claim 20, wherein the server is further operable to store messages that have a same priority level and were received from different clients in order of the different clients' rotation positions.

23. The network system of Claim 16, wherein the server is further operable to receive a first message from a first client and a second message from a second client, to store the first message before the second message in a queue if the first message's priority level is higher than the second message's priority level, to store the first message before the second message in the queue if the first and second messages have the same priority level and the first client's rotation position is before the second client's rotation position, and to process the first and second message in order of storage in the queue.

24. A server operable to couple to a plurality of clients, to receive one or more messages requesting processing by the server from the clients, to determine a priority level for each message, and to process the messages according to the messages' priority levels and the clients' rotation positions.

25. The server of Claim 24, wherein the server is further operable to process messages that have different priority levels in order of the different priority levels.

26. The server of Claim 24, wherein the server is further operable to process messages that have a same priority level and were received from different clients in order of the different clients' rotation positions.

27. The server of Claim 24, wherein the server is further operable to receive a first message from a first client and a second message from a second client, to process the first message before the second message if the first message's priority level is higher than the second message's priority level, and to process the first message before the second message if the first and second messages have the same priority level and the first client's rotation position is before the second client's rotation position.

28. The server of Claim 24, wherein the server is further operable to store the messages in a queue according to the messages' priority levels and the clients' rotation positions and to process the messages in order of storage in the queue.

29. The server of Claim 28, where the server is further operable to store messages that have different priority levels in order of the different priority levels.

30. The server of Claim 28, where the server is further operable to store messages that have a same priority level and were received from different clients in order of the different clients' rotation positions.

31. The server of Claim 24, wherein the server is further operable to receive a first message from a first client and a second message from a second client, to store the first message before the second message in a queue if the first message's priority level is higher than the second message's priority level, to store the first message before the second message in the queue if the first and second messages have the same priority level and the first client's rotation position is before the second client's rotation position, and to process the first and second message in order of storage in the queue.

32. A method for processing messages at a server, the method comprising:

receiving a first message from a first client, the first message requesting processing by the server;

determining the first message's priority level;

receiving a second message from a second client, the second message requesting processing by the server;

determining the second message's priority level; and

processing the messages in order according to the messages' priority levels and the clients' rotation positions.

33. The method of Claim 32, wherein processing the messages in order according to the messages' priority levels and the clients' rotation positions further comprises:

processing the messages in order of the messages' priority levels if the messages have different priority levels; and

processing the messages in order of the clients' rotation positions if the messages have a same priority level.

34. The method of Claim 32, wherein processing the messages in order according to the messages' priority levels and the clients' rotation positions further comprises:

processing the first message before the second message if the first message's priority level is higher than the second message's priority level; and

processing the first message before the second message if the first and second messages have a same priority level and the first client's rotation position is before the second client's rotation position.

35. The method of Claim 32, wherein processing the messages in order according to the messages' priority levels and the clients' rotation positions further comprises:

storing the messages in a queue in order of the messages' priority levels if the messages have different priority levels;

storing the messages in the queue in order of the clients' rotation positions if the messages have a same priority level; and

processing the messages in order of storage in the queue.

36. The method of Claim 32, wherein processing the messages in order according to the messages' priority levels and the clients' rotation positions further comprises:

storing the first message before the second message in a queue if the first message's priority level is higher than the second message's priority level;

storing the first message before the second message in the queue if the first and second messages have a same priority level and the first client's rotation position is before the second client's rotation position; and

processing the first and second messages in order of storage in the queue.